PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04N 5/262

A1

(11) International Publication Number: WO 98/32281

(43) International Publication Date: 23 July 1998 (23.07.98)

(21) International Application Number: PCT/US98/01036

(22) International Filing Date: 21 January 1998 (21.01.98)

(30) Priority Data:

60/035,975	21 January 1997 (21.01.97)	US
08/996,871	23 December 1997 (23.12.97)	US

(71) Applicant: SARNOFF CORPORATION [US/US]; 201 Washington Road, CN 5300, Princeton, NJ 08543 (US).

(72) Inventor: HURST, Robert, Norman, Jr.; 68 Hart Avenue, Hopewell, NJ 08525 (US).

(74) Agents: SILVERIO, John, V. et al.; Sarnoff Corporation, 201 Washington Road, CN 5300, Princeton, NJ 08543 (US).

(81) Designated States: BR, CA, CN, JP, KR, MX, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

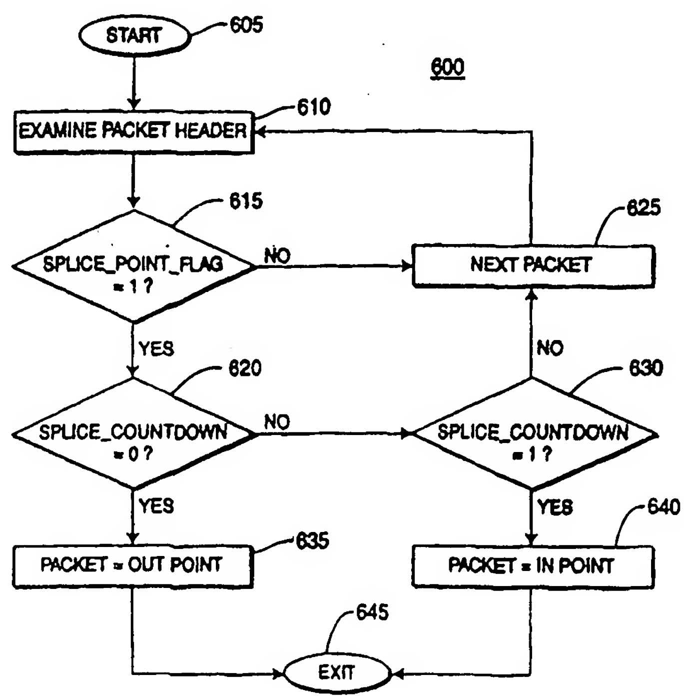
Published

With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: INFORMATION STREAM SYNTAX FOR INDICATING THE PRESENCE OF A SPLICE POINT

(57) Abstract

A method (600) for identifying splicing in-points and splicing out-points in, illustratively, a bitstream. The method (600) examines (610) header information in the bitstream and, in response to valid splice countdown information comprising a first (630) or second (620) value, identifies a packet as, respectively, an in-point or an out-point.



Information Stream Syntax for Indicating the Presence of a Splice Point

This invention was made with U.S. government support under contract number 70NANB5H1174. The U.S. Government has certain rights in this invention.

This application is a continuation in part of U.S. Patent Application Serial No. 08/864,322, filed on May 28, 1997 for a METHOD AND APPARATUS FOR SPLICING COMPRESSED INFORMATION STREAMS, which is herein incorporated by reference.

The invention relates to communication systems in general, and more particularly, the invention relates to a method for identifying and utilizing splicing “in-points” and splicing “out-points” in MPEG-like information stream.

BACKGROUND OF THE DISCLOSURE

In several communications systems, the data to be transmitted is compressed so that the available bandwidth is used more efficiently. For example, the Moving Pictures Experts Group (MPEG) has promulgated several standards relating to digital data delivery systems. The first, known as MPEG-1 refers to ISO/IEC standards 11172, incorporated herein by reference. The second, known as MPEG-2, refers to ISO/IEC standards 13818, incorporated herein by reference. A compressed digital video system is described in the Advanced Television Systems Committee (ATSC) digital television standard document A/53, incorporated herein by reference.

A program transport stream is formed by multiplexing individual elementary streams which share a common time base (i.e., the same 27MHz clock source). The elementary streams comprise encoded video, audio or other bit streams. The elementary streams may be, but do not have to be, in a packetized elementary stream (PES) format prior to transport multiplexing. A PES consists of a packet header followed by a packet payload. As the elementary streams are multiplexed, they are formed into transport packets and a control bit stream that describes the program (also formed into transport packets) is added.

There are many instances where there is a need to switch from one encoded or compressed bitstream to another. When switching from one compressed MPEG

The teachings of the present invention can be readily understood by considering the following detailed description in conjunction with the accompanying drawings, in which:

FIG. 1 shows a block diagram of a compressed bitstream splicing system including the invention;

FIG. 2 depicts a flow chart of a seamless splicing process in accordance with the invention;

FIG. 3 shows a detailed block diagram of the splicer of FIG. 1;

FIG. 4 depicts a block diagram of digital studio comprising a plurality of interoperable islands and including the invention;

FIGs. 5A-5C depicts a plurality of splicing scenarios; and

FIG. 6A and FIG. 6B together depict a flow diagram of a routine 600 suitable for identifying in-points and out-points in accordance with the invention.

To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures.

DETAILED DESCRIPTION

The invention is generally described within the context of a digital television studio includes a plurality of operative environments which receive and process various bitstreams and which have associated switching capabilities according to the invention. The switching capabilities allow seamless or non-seamless splicing of a plurality of, e.g., video transport streams to produce an output stream. A combination of seamless and non-seamless bitstreams may be produced to provide a controllably degraded output stream.

The invention is a two-input bitstream splicer which performs switching, splicing or insertion operations on a pair of MPEG-compliant input transport streams to produce an output stream. It must be noted that the principles of the invention apply to bitstream switchers or splicers having more than two inputs and to input streams other than MPEG-compliant input streams. The invention may be implemented using a general purpose computer system that is programmed to perform the functions discussed below. As programmed, the general purpose computer becomes a specific purpose apparatus for splicing digital data bit streams.

end of the last video transport packet of the stream of interest. The video stream before and through the last packet must meet the splicing definition of an out-point. Similarly, a video stream in-point is the beginning of the first video transport packet of a splice segment (SS). It must be noted that other information in the transport stream, specifically audio, is unlikely to be neatly segmented at in-points and out-points. A method for correcting errors induced by the non-alignment of audio transport packets is described in U.S. patent application serial number 08/864,321, filed May 28, 1997, incorporated herein by reference.

A critical aspect of splicing information streams is the proper processing of the various delay parameters. One parameter of concern is the delay parameter associated with the various information streams. In the case of an MPEG-compliant stream, the delay parameter is the video buffering verifier (VBV) delay parameter. Another parameter is the latency, or transitional period, inherent in a splicing operation. For example, a typical splice occurs at a certain time, i.e., a "splice time." Prior to the splice time an output information stream comprises a from-stream. At the splice time, a switch to the to-stream occurs. For a period of time the output stream may include information from both the from-stream and the to-stream. Eventually the output stream includes information from only the to-stream.

It is assumed that the from-stream and the to-stream are each valid. There are certain constraints on the streams that must be met if the splicing is to be seamless. Seamless splicing implies that the resultant spliced bitstream will not cause discontinuities in the future.

One specific example of a valid splice segment that can be seamlessly spliced is an MPEG-compliant splice segment. An MPEG Splice Segment (SS) is defined at the transport level and includes functionality at the video (and audio) levels. An information-bearing splice segment may be as short as a single frame. A splice segment may even be a zero frame length segment (although such a SS might be MPEG non-compliant). Such a zero-length segment is simply an in-point followed by an out-point (i.e., an "in-out-point"). A SS may be also be very long, including many GOPs. In general the length of a SS is not constrained and the SS should include multiple out-points to enable seamless exiting from the

a video disk, tape machine, or other storage device) which stores video and audio elementary streams and transport encodes the stored streams to produce a second MPEG-compliant transport stream S7. The stored information may comprise, e.g., advertisement or local programming information to be spliced into the first transport stream. The splicer 300 selectively couples one of the two input transport streams S6, S7 to a transmitter or other subsystem as an output stream S9. An optional splice monitor 130 monitors various parameters of the spliced output signal S9, e.g., delay parameter, buffer utilization information, synchronization, bitstream source and the like. The optional splice monitor 130 is responsive to the controller 105 and the splicer 300.

The splicer 300 receives the first transport stream S6, illustratively a television program produced by a first source, and the second transport stream S7, illustratively an advertisement produced by a second source. In response to a control signal SELECT, the splicer produces an output signal S9 comprising either the first S6 or second S7 transport stream. The control signal SELECT may include priority information which causes the splicer 300 to respond immediately, within a defined time interval or when certain conditions exist (i.e., specific alignments of stream entrance or exit points). The splicer 300 produces a signal ACKNOWLEDGE which is used to acknowledge the SELECT signal and provide specific details about the splice operation (e.g., exact time of splice, error conditions and the like). The operation of the splicer 300 is described more fully below with respect to FIG. 3.

The actual splicing operation is the process that takes place within the splicer 300 that does what is necessary to actually switch amongst the bitstreams. This involves stopping, in an orderly manner, the flow of packets from the from-stream; starting, in an orderly manner, the flow of packets from the to-stream; and adjusting the header information in the output stream. During some interval, packets from both the from-stream and the to-stream are likely to be intermixed.

Splicing operations must be synchronized to be seamless. To ensure that input streams arrive at the appropriate splicers at the time they are needed several synchronizing operations may be performed. It is assumed that the

The decision to splice may be content related, such as a switch from a from-stream to a to-stream when a content-related data element is encountered in one of the streams. For example, the from-stream may be monitored and, in response to the detection of, e.g., a black-screen or a scene change, a splice decision may be made. This operational decision does not require synchronization. Rather, the decision requires that the splicer (or a controller) analyze, e.g., the from-stream to detect the data element. The decision to splice may also be data-flow related, such as a switch from a from-stream to a to-stream on some particular packet or upon the start or stop of information flow.

The decision to splice may be time-related, such as a switch from a program to commercial at noon. Time-related decisions must be referenced to the splicer's local frame-of-reference. A message-passing process passes the decision information to the splicer in time for the splicer to be ready to make the splice in its frame-of-reference. Given that the decision to splice at some time has been made, the splice will be made at the next available splice point, based upon the from-stream and the to-stream.

The decision to splice may be may be event driven, such as the pushing of a button (e.g., the director's "take" command, as depicted in the splicer 100 of FIG. 1). When the message indicating the event arrives at the splicer, the actions are the same as those for a time-related decision whose time has arrived.

Some form of acknowledge message may be required. This message, when delivered to the originator of the splice decision (e.g., the controller), will allow an intelligent choice to be made about time-outs, and actions like panic non-seamless splices. Time-outs and determinations about corrective actions to remedy splice failures is a policy matter for the originator of the splice decision. Time-out and forced switch may be a service implemented by the splicer but only as a convenience.

An operational unit (e.g., splicer or switcher) may feed back an appropriate acknowledgment message to a controlling entity. The contents of such a feedback message may include one or more of the following parameters: 1) a splice did or did not take place; 2) the local time-of-day that the splice occurred; 3) the delay-parameter value of the to-stream; 4) the delay-parameter value of the

are available when out-points occur in the from-stream. If the amount buffered is insufficient (e.g., more than a second elapses between successive in-points in a from-stream), then the buffer will overflow and will contain invalid information. This condition is remedied by an appropriate number of in-points and out-points being inserted into the bitstreams. If bitstreams do not have in-points and out-points often enough, then those bitstreams can not be seamlessly spliced at those times. Moreover, to the extent that there is packet or cell jitter in the arrival time of input bitstreams, a first-in, first-out (FIFO) buffer (with output clocked at nominal data rate) is expected to smooth the flow.

The synchronization of server generated streams will now be discussed. Server-generated streams must be carefully generated so that the data does not arrive at the splicer too early or too late. If the data arrives too early, there is some risk of overflow of an input buffer. If it is assumed that the splicer has enough synchronization buffering to hold a second or so of video, then it would seem that server streams can be delivered in any pattern of flow that never exceeds the just-in-time limit, and the one-second-early limit. Of course, there may be peak rate limitations on the splicer.

The synchronization of remotely-generated streams will now be discussed. It must be noted that any stream processed in a studio containing the splicer is expected to have the same reference clock rate. Remotely-generated streams, by the time they have reached a splicer, should be the same as locally-generated real-time streams. To reference remotely-generated streams to a local master clock the remote source may be genlocked to the local studio. This can be done via a reverse channel or by locking both to an external reference, such as a timing signal derived form the Global Positioning System (GPS). If there are two independent studios, each with an independent master clock, and each is doing a remote feed to the other, then one will be delivering data too slow for the other, and one will be delivering data too fast. Another method is to delay a remote feed by a time equal to the maximum clock drift over some operating interval. A 30ppm. drift rate, over 24 hours, accumulates 2.6 seconds. A six second buffer, initialized to a 3 second fill, is adequate to absorb clock drift.

and the splice will be made. It must be noted that the from-stream contains up to 1/4 second of delay. One monitor delay later (1/2 second) the scene on the output monitor changes.

If the director responded to a scene on the output monitor 132, the amount of output monitor delay (i.e., the time between the "take" command TAKE and a change in scene on the output monitor 132) is between 1/2 and one second. If the director responded to a scene on the from-stream monitor 136 , the amount of from-stream monitor delay is between 1/4 and 1/2 second and the output monitor delay is 1/2 second. If the director responded to a scene on the to-stream monitor 134, the to-stream monitor 134 is continuous (i.e., no monitor delay) and the output monitor delay is negative 1/4 seconds (i.e., the scene changes 1/4 second after the "take" button TAKE is pressed and the image displayed occurred 1/4 second prior to the press of the button).

A second example is the "next" mode of operation. In this mode, a queued up to-stream is flushed from a to-stream synchronization buffer and the next segment beginning with an in-point is queued up within up to 1/4 second. The to-stream synchronization buffer also has zero to 1/4 seconds of random delay. When the in-point arrives the splice is made.

If the director responded to a scene on the output monitor 132, the amount of output monitor delay is between 1/2 and one second. If the director responded to a scene on the from-stream monitor 136, the amount of from-stream monitor delay is between 1/2 and 3/4 second and the output monitor delay is 1/2 second. If the director responded to a scene on the to-stream monitor 134, the to-stream monitor 134 is continuous and the output monitor 132 switches to a new scene between zero and 1/4 second later.

The choice of "soonest" or "next" mode of splicing is an operational one, and may be based upon which disconcerting effect (delay or back-up) is least objectionable. To alleviate these effects an amount of delay may be inserted into the splicer inputs. If this delay matches the monitor delay, and the monitors are connected to the inputs of the delays, then the apparent delay between monitor scenes and button action is less, but the delay to final output is greater. In addition, a separate monitor control unit may be built to simulate the bit-stream

buffer 320A. The second bitstream examiner 310A examines the second bitstream for exit points which have been included in the second input bitstream S6. In the "selected mode" of operation, the second bitstream examiner 310B is not used and the second synchronization buffer 320B serves as a constant delay buffer which produces a delayed bitstream S3B.

The delayed bitstream S3B is coupled to a working buffer 330B and a switch controller 340. The second working buffer 330B produces an output signal S4B which is coupled to packet switching unit 350. The second working buffer 330B holds the selected bitstream long enough to allow for overlap of old audio packets with current video packets. This allows audio frames to continue to completion after a splice is made. The synchronization of audio and video frames are discussed in more detail below and in U.S. patent application serial number 08/864,321, filed May 28, 1997 and incorporated herein by reference.

A splice decision is made by a controller (e.g., controller 105) and coupled to the switch controller 340 via a control signal SELECT. Assuming that the splice decision equates to the command "splice seamlessly at the next opportunity," the switch controller 340 responds by scanning the currently selected output stream (i.e., bitstream S3B) for out-points. It is assumed that an in-point is positioned at the end of the first synchronization buffer 320A. When an out-point arrives on the from-stream, the switch controller 340 causes, via a control signal A/B, the switch 350 to begin coupling video packets from the to-stream through the switch to an optional header adjuster. At an appropriate time any audio packets within the to-stream are also switched.

The optional header adjuster 360 alters time-stamps in the selected output stream S8 to produce a retimed output stream S9. The retiming of the program clock reference (PCR), presentation time stamps (PTS) and decode time stamps (DTS) of the selected stream S8 may be necessary to ensure that the splice is, in fact, seamless to a decoder. The header adjuster 360 includes a 27MHz (local) station clock 362 which is utilized by a local PCR and PCRB generator 364. To retime the presentation and decode time stamps it is necessary to partially decode (i.e., packetized elementary stream (PES) layer) the selected transport stream S8. The partial decoding and retiming of the PTS and DTS is performed by a PTS and

requirements. If the splice decision is made in the context of creating a live production, and the production involves information stored on servers, it is helpful to know that anticipated splice-points are about to arrive soon (via, e.g., splice point countdowns or splicing tables). The splicing decision and related matters will be discussed in detail below.

The invention will now be described within the context of a digital television studio including a number of distinct operating environments (such as servers or edit-suites) which receive, process and transmit various information streams. The operating environments, or "islands of interoperability," may be interconnected to perform one or more operations on the various information streams. The studio output may be delivered to end-users (e.g., the public) via ATSC broadcast, cable, telephone and satellite transmission and the like. The studio output may also be stored for later use in, e.g., a server or on CD-ROM or video tape. The invention is also useful in video teleconferencing and other applications.

While the streams delivered to broadcast customers must meet, e.g., ATSC standards, it is not necessary to deliver all of the internal studio information. For example, high bit-rate studio formats are useful only within the studio or studio-like environments. When dealing with splicing, there may be information within the stream that is meaningless to consumer decoders, but which is necessary to studio splicing.

FIG. 4 depicts a block diagram of digital studio comprising a plurality of interoperable islands and including the invention. The digital studio 400 of FIG. 4 includes interoperable islands 401, 402 and 404-409. The digital studio 400 also includes a first compressed bitstream stream source 110, a second compressed bitstream stream source 120, a splicer 300, a controller 105 and an optional splice monitoring unit. The first compressed bitstream stream source 110, illustratively a "live feed" from a transport stream encoder, produces a first MPEG-compliant transport stream S6. The second compressed bitstream stream source 120, illustratively a server (e.g., a video disk, tape machine, or other storage device) which stores video and audio elementary streams and transport encodes the stores streams to produce a second MPEG-compliant transport stream S7. The

An alternate mode of studio operation is to controllably operate one or more islands in a non-seamless mode. The non-seamless mode may be required in several circumstances where a splice or other transition between bitstreams must occur rapidly, and a range of bitstream degradation is permissible. It must be noted that non-seamless switching may produce errors which are propagated to subsequent islands receiving a degraded bitstream. These errors may be mitigated, if necessary, by, e.g., dropping damaged or inferior access units or groups of access units (e.g., video frames) or by adding additional access units. For example, if a to-stream having a short delay-parameter is to be spliced onto a from-stream with a long delay-parameter, the splicing operation is unlikely to be seamless (i.e., the buffer will likely overflow). In this case, frames may be dropped to avoid the overflow condition. Also, when a to-stream with a long delay-parameter is to be spliced onto a from-stream with a short delay-parameter, the splicer needs to adjust time stamps to cause a number of frame repeats (i.e., add frames) while the buffer fills. The buffer may also be increased by splicing short, all-black frames on the end of a short delay-parameter sequence to build up the value of the delay-parameter in current use.

In the exemplary embodiments of FIGs. 1, 3 and 4, splicing operations take place in operational units (e.g., splicing islands), such as routing switchers, play-to-air switchers, production switchers or other switchers. Therefore, it is desirable to support a plurality of data formats and bitrates. For example, the so-called 422@HIGH and 420@HIGH television studio formats each support multiple picture formats and bit rates. Therefore, it may be necessary to splice, e.g., a bitstream comprising a 1280 by 960 picture element, 60Hz Progressive Scan picture onto the end of a bitstream comprising a 1920 by 1080 picture element, 59.94Hz interlaced picture. Moreover, it may also be necessary to splice a 45 Mb/s stream onto the end of a 155 Mb/s stream.

Both of the above example splices may be seamlessly made if the streams being spliced have matching delay parameters. Therefore, it is important that the controller that makes the splice decision know the delay parameters of the various streams to be spliced. The delay parameter of a stream may be calculated by an operational unit receiving a stream or included within the stream as part of the

associated with an information stream as the information stream is processed, and the transmission or regeneration of the table as the information stream is transmitted (e.g., via satellite link). It should be noted that the use of separate information tables to determine splice locations is practical within a server or island of operability, though this use of tables may be less practical when transmission encoding and decoding are to be performed on the stream.

Third, in-point and out-point markers may be placed within the information stream directly. An MPEG compliant information stream includes header portions where such a marker may be included. There are header portions suitable for in-point and out-point marker insertion at the system level, transport level and PES level. There are also opportunities to insert markers in the elementary streams.

Both in-points and out-point should be marked and, ideally, the marking should occur at the system, transport and PES levels. In addition to the insertion of in-point and out-point markers, the delay-parameter associated with the stream or splicing segment and an audio offset (i.e., a displacement of audio-frame boundaries from associated video frames) should also be inserted into one or more layers of the information stream. The MPEG count-down feature should also be used to indicate that, e.g., an out-point is approaching (decreasingly positive countdown) or an in-point has been transmitted (increasingly negative-countdown). For various business reasons it may be desirable to remove these markers prior to transmitting an MPEG or ATSC signal to an end-user (i.e., consumer). While end-users might want to splice the video, it is important that they not be able to clip out commercials automatically.

The above-described redundancy of marking provides a maximum flexibility to system designers and provides a redundancy of operation which helps to ensure that splicing operations are, in fact, seamlessly made (i.e., made at appropriate in-points and out-points).

Bitstream Generation

To help ensure seamless splicing it may be necessary to create the bitstreams to be spliced in a certain manner. There are two facets to the creation of bitstreams that can be spliced; the creation of the stream content, and the

decoder buffer does not increase from in-point to out-point. It must be noted that so-called "stuffing bits" are not counted, since these bits are only included to meet specified transmission rates and disappear from the buffer when the real bits are used (i.e., stuffing bits do not accumulate in the buffer).

Another rate control issue involves the presentation time of decoded information frames in relation to the time the buffer receives the next frame. For example, the decoder buffer will not underflow if, at an out-point, the time (measured in time units) to display the not-yet displayed buffer contents is greater than the time (measured in time units) for the first I-frame to be delivered to the decoder at the specified bit rate. In other words, the next I-frame (the first frame of the to-stream) must be delivered to the decoder buffer before the buffer is emptied.

The above-described amount of time may be defined at the "Delay-Parameter" for the stream. The frame sizes (measured in time to transmit the frames at the operating bit rate) must be consistent with the operating delay parameters to ensure seamless splicing. The delay parameters are the end-to-end VBV size (measured in time) and the VBV contents (measured in time) at the beginning/end of a stream. An additional, globally defined value is the maximum size of physical buffers (in bits). This maximum size must be greater than the maximum VBV size implied by the MPEG profile and level indication criteria. Finally, the decode time stamp at an out-point of a from-stream should be one frame time of the stream greater that the DTS and PTS of the last frame of the from-stream.

As previously mentioned, it is important to distinguish between a splicing decision and the actual process of splicing. A splicing decision is made by some human. The decision may be made in the process of generating a list of programming to be transmitted by a television studio or in real time as the studio is transmitting. The splicing decision may be made by some surrogate process, such as a preprogrammed command to splice a station identification announcement into the studio transmission every day at 12:05 AM. The decision may be to splice at a particular time in the future or immediately.

Illustrative Examples of Video Splicing

The above-described concepts will now be illustrated using several examples of MPEG bitstream splicing operations. The first example is an all I-frame low delay splicing example. A to-stream comprises either 24 or 30 frames per second (fps) video streams including only I-frames. The delay parameter of the to-stream is equal to one frame time at the slowest frame rate (i.e., 42mS if 24 fps). In this example each of the I-frames contains fewer bits than can be sent in one unit of display time (i.e., one 42mS frame time) at the bit rate for the frame. If the bit rate is 150 Mb/s, a 30fps frame contains no more than 5 Mb. If the bit rate is 150 Mb/s, a 24fps frame contains no more than 6.25 Mb.

When the last bit of a from-stream having the same parameters has entered a decoder buffer, the presentation time stamp indicating when the last frame is to be presented has a value 42mS in the future. Thus, if the from-stream were at 30Hz. (33mS frame rate), 9 ms. after the out-point the last frame of the from stream will be taken from the decoder buffer, and 33mS later the first frame of the to-stream will be needed. If the to-stream is also 30Hz, the first frame will have been delivered 9mS before it is needed. If the from-stream were at 24Hz, and the to-stream were also at 24Hz, the to-stream frame arrives just in time. If the from-stream were at 60Hz, (17mS) then when the last bit of the from-stream is delivered, the decoder buffer contains 2 frames (33mS) and the decoder will not use the first of them for 9mS. If the bit-rates of the from and to sequences differ, as long as the bits are delivered at the rate corresponding to the sequence bit-rate, the time to load frames remains correct. To summarize, streams are coded with a bit-count between the in-point and following out-points that is calculated from the bit-rate and the frame-time (i.e., bit-rate * frame-time). The presentation time-stamps are set to values that all agree with the delay-parameter (i.e., first frame presented delay-parameter after the first bit arrives.).

The second example is a complex GOP transmission format. For purposes of the second example, it is assumed that stream is a 30 frame per second video (and associated audio) stream having delay-parameter of 250mS, a display order of "...IBBPBBPBBPBBPBBP..."and transmission order of "...IPBBPBBPBBPBBPBB..." (where "I" represents an I-frame, "P" represents a

delay in the buffer is approximately zero. Each P-frame now adds 400 Kb in 20mS and every 33 ms 400Kb is used. Thus, delay in buffer increases by 13mS every frame time. After 15 frames, the delay stored in the buffer has reached the delay-parameter value. At this time a splice to another sequence may be made because the buffer is able to receive an I-frame.

Compressed Audio Splicing

The following discussion of splicing of compressed audio is limited to the issue of splicing combined video-audio streams in the audio-follows-video mode. The composition of streams from separately edited audio and video streams is not considered here.

Compressed audio is carried in frames. Each audio frame is of fixed duration and contains a fixed number of bits. Unfortunately, the audio frame size, or duration, is different from any of the video frame sizes, or durations. This means that audio frames will not align with splice points. Audio frames can be considered to be randomly aligned with the video. Therefore, when making a splice, the alignment of the audio with the video will be different for the to-stream and the from-stream.

It is desirable to ensure alignment of audio information to corresponding video information (i.e., "lip-sync"). That is, the audio and video must remain properly phased with respect to each other. A Presentation Time Stamp (PTS) exists in each audio stream. The audio and video PTSs refer to the same reference to allow the required synchronization. When the splice is complete, the to-stream becomes the output stream. It is important to note that, particularly due to audio constraints, the process of switching may extend in time before and after the actual switch instant.

Audio information frames in an information stream are ideally located within a limited time difference from respective video information arrival at the end of the decoder buffer. If there is a 1/2 second end-to-end video buffer delay, then audio packets should be approximately 1/2 second later in a transmission stream than corresponding (i.e., having the same presentation time stamps) video packets. If this assumption is correct, then the switching operational unit must save audio information from the from stream for this 1/2 second after the video

FIG. 5 depicts a plurality of splicing scenarios involving audio alignment which illustrate aspects of audio-video splicing, assuming the above cited third approach is used to maintain alignment of audio with its corresponding video.

FIG. 5A depicts the simple splicing case where both audio streams align with their corresponding video. The splicer delays both to-streams and simply switches at the splice point.

FIG. 5B depicts the splicing case where the from-stream video and from-stream audio are aligned, but the to-stream video and to-stream audio are not aligned. When the splice is made, a partial to-stream audio frame is discarded. The next complete to-audio frame is passed to the output with appropriate delay.

FIG. 5C depicts the typical splicing case where both audio streams do not align with the corresponding video streams. It can be seen that a from-stream audio frame has already begun before the splice point. This audio-frame is buffered and transferred to the output. It doesn't end until some fraction of a frame time after the splice. The to-stream audio frame that spans the splice-point cannot be used. The next to-stream audio frame also cannot be used. It begins too early, and would overlap the last from-stream audio frame. The first to-stream audio frame that appears in the output stream begins D time units after the splice-point. This delay may be as much as two audio frames.

In the above example (FIG. 5C) the lip-sync is preserved, but as much as 32mS of from-stream audio overlaps the to-stream video. Also, the first to-stream audio begins as late as 64mS after to-stream video begins. Finally, the splicer performing the splicing operation must buffer a whole audio frame in each work buffer.

Ideally, every audio frame includes a PTS. It is possible that some equipment manufacturers only include a PTS every, e.g., third audio frame. In this case, or the case where there is no audio PTS, a splicing operation may be performed after calculating a virtual time-stamp. The virtual time stamp is derived from the approximate real-time delay of audio-frames from video reference time-stamps. The virtual audio-time-stamp is then incremented by the (known) audio-frame duration on successive audio-frame starts. This process may

a 4-bit field used to derive splice_decoding_delay and max_splice_rate data from, e.g., a table storing such data. The standard use the these header flags and fields to implement a splicing function is defined in the MPEG specification.

As previously discussed, splice points within a transport stream may be in-points, out-points or both. An out-point is equivalent to the MPEG-definition of a splice point. An in-point comprises a splice point (i.e., an out-point) followed by a sequence header that is immediately followed by an I-frame. Therefore, in-points within a particular stream may be identified by finding out-points followed by sequence headers that are followed immediately by I-frames. Thus, a to-stream may be entered at any in-point, as described above, even if the out-point of a from-stream is not followed by a sequence header or an I-frame. In this embodiment, the splicing_point_flag of the packet immediately preceding the out-point must equal one, and the splice_countdown field of that packet must equal zero. However, the above-described embodiment requires that the bitstream be parsed down to the elementary layer to examine, e.g., the picture_coding_type field in the picture header to determine if an I-frame is present.

In the preferred embodiment of the invention, entrance and exit indicia comprise information residing within the transport layer, thereby obviating the need to parse the bitstream down to its elementary layer. Specifically, an out-point in a from-stream is indicated by the splicing_point_flag being equal to one and the splice_countdown field being equal to zero. Similarly, an in-point in a to-stream is indicated by the splicing_point_flag being equal to one and the splice_countdown field being equal to negative one. Thus, an in-point packet (i.e., the packet that immediately follows an in-point) and an out-point packet (i.e., the packet including an out-point) may be the same packet. Moreover, the splice_type of an in-point packet (i.e., the packet that immediately follows an in-point) indicates the suitability of splicing the in-packet to an out-packet in that the in-packet and the out-packet should both have the same splice_type.

In this embodiment, an out-point that is not also an in-point must have the splicing_point_flag equal to zero in the packet immediately following the packet with the splicing_point_flag being equal to one and the splice_countdown being equal to zero. In this manner, the splicing_point_flag indicates that the

and 310B of the splicer 300 depicted in FIG. 3. The routine 600 is entered at step 605 when a transport packet within a stream to be examined (e.g., S6 or S7) is received by, e.g., a bitstream examiner (e.g., 310A or 310B). The routine 600 proceeds to step 610, where the packet header of the received packet is examined, and to step 615, where a query is made as to whether the splice_point_flag within the adaptation header of the received packet is equal to 1.

If the query at step 615 is answered affirmatively, the routine 600 proceeds to step 620, where a query is made as to whether the splice_countdown flag is equal to 0. If the query at step 620 is answered affirmatively, then the routine proceeds to step 635, where the packet is identified as containing an out-point. Such identification may take the form of setting an "out-point-ready" flag suitable for use in, e.g., step 210 of the routine 200 of FIG. 2. The routine then proceeds to step 635, where it is exited.

If the query at step 620 is answered negatively, then the routine 600 proceeds to step 630, where a query is made as to whether the splice_countdown flag is equal to -1. If the query at step 630 is answered affirmatively, then the routine proceeds to step 640, where the packet is identified as containing an in-point. Such identification may take the form of setting an "in-point-queued" flag suitable for use in step 208 of the routine 200 of FIG. 2. The routine then proceeds to step 645, where it is exited.

Although various embodiments which incorporate the teachings of the present invention have been shown and described in detail herein, those skilled in the art can readily devise many other varied embodiments that still incorporate these teachings.

setting to a second value a splice countdown field within said header portion of said identified packets to include said entrance indicium, and setting to a third value said splice countdown field within said header portion of identified packets to include said exit indicium.

5. The method of claim 4, wherein said first value is one, said second value is negative one, and said third value is zero.

6. The method of claim 4, wherein said entrance indicium is associated with a packet the precedes a sequence header that immediately precedes, in the case of a video stream, an I-frame.

7. A method for splicing bitstreams, said bitstreams comprising a plurality of information segments representative of a sequence of information frames, said method comprising the steps of:

monitoring (210) a first bitstream to detect an exit indicium, said first bitstream being coupled to an output and including at least one of said exit indicium, said exit indicium indicative of an appropriate last information segment of a splicing segment;

providing a second bitstream, said second bitstream including at least one entrance indicium, said entrance indicium indicative of an appropriate first information segment of a splicing segment; and

coupling (220) said second bitstream to said output in response to a control signal, a detection (620) of said exit indicium in said first bitstream and a detection (630) of said entrance indicium in said second bitstream; wherein

said exit indicium comprises a valid splice countdown field in said header portion of said information segment being set to a first value.

8. The method of claim 7, wherein said first value is zero.

14. The apparatus of claim 11, wherein said bitstream examiner generates an output signal (S2A; S2B) indicative of said one of said entrance indicium and said exit indicium.

15. The apparatus of claim 14, further comprising:

a utilization circuit (320A; 320B), coupled to said bitstream examiner, for processing said bitstream in response to said bitstream examiner output signal.

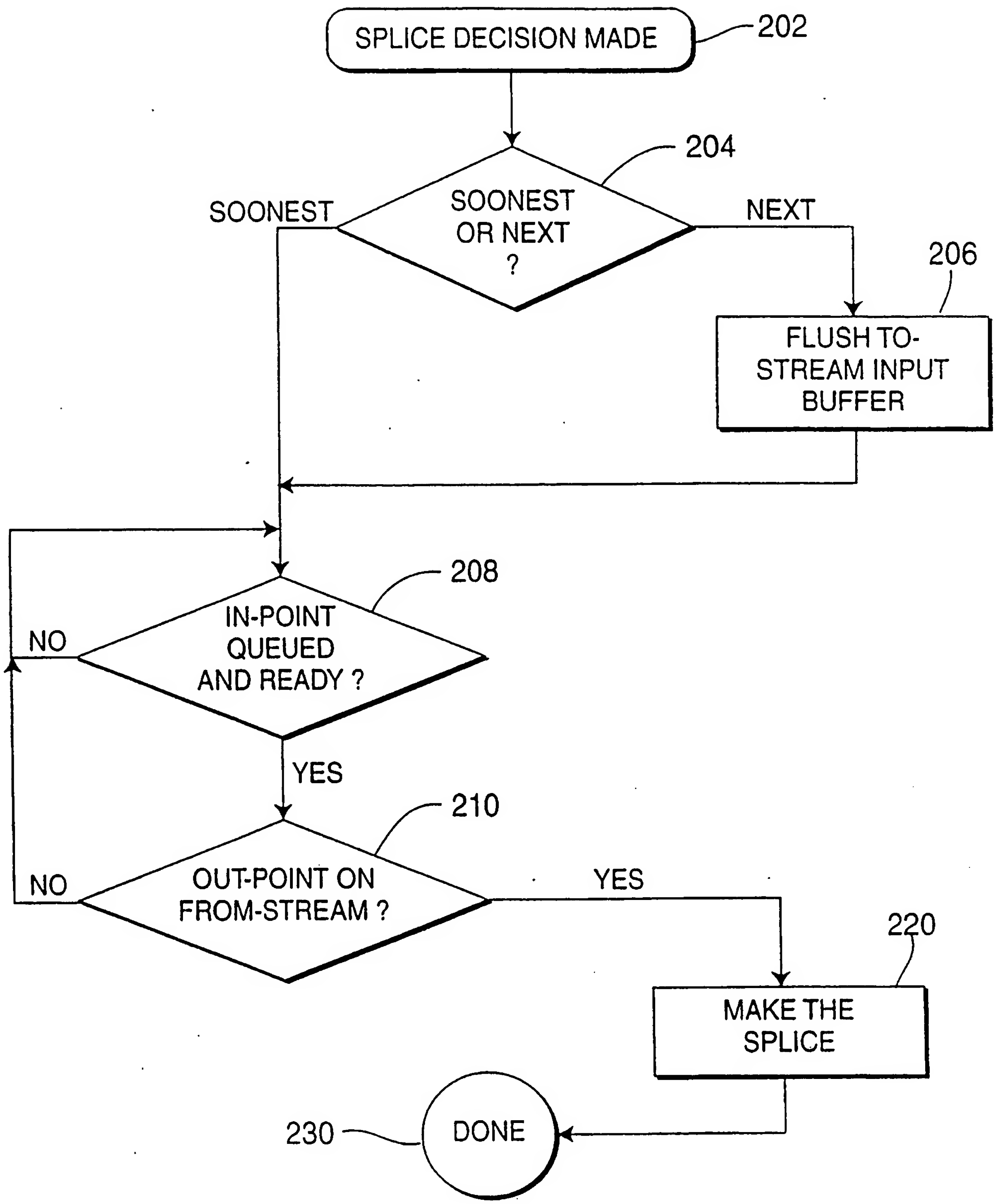

SPLICE DECISION MADE
202
SOONEST OR NEXT ?
204
SOONEST
NEXT
206
FLUSH TO-STREAM INPUT BUFFER
208
IN-POINT QUEUED AND READY ?
NO
YES
210
OUT-POINT ON FROM-STREAM ?
NO
YES
220
MAKE THE SPLICE
230
DONE

FIG. 2

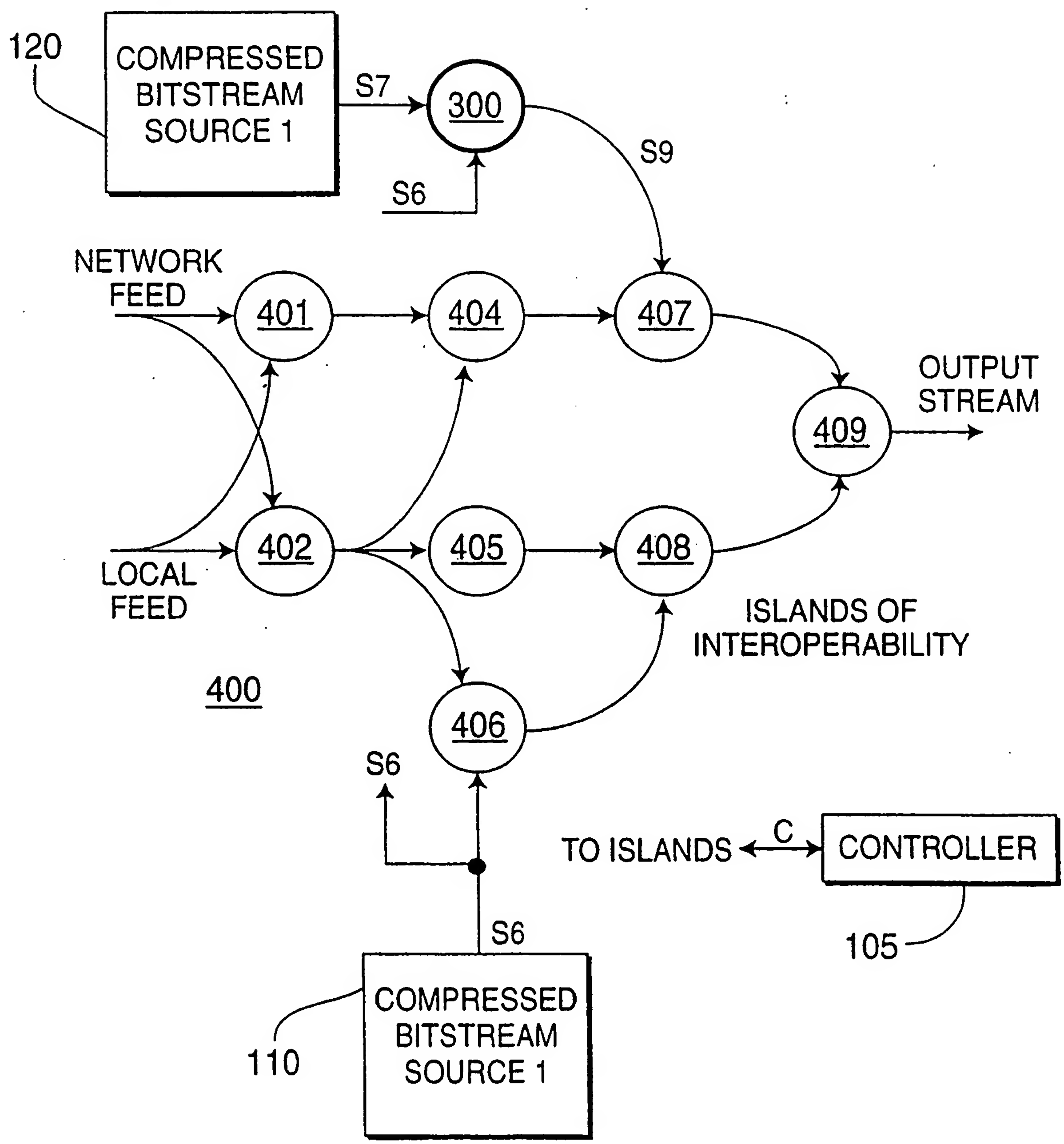

120
COMPRESSED BITSTREAM SOURCE 1
S7
300
S6
S9
NETWORK FEED
401
404
407
OUTPUT STREAM
409
LOCAL FEED
402
405
408
ISLANDS OF INTEROPERABILITY
400
406
S6
TO ISLANDS
C
CONTROLLER
S6
105
110
COMPRESSED BITSTREAM SOURCE 1

FIG. 4

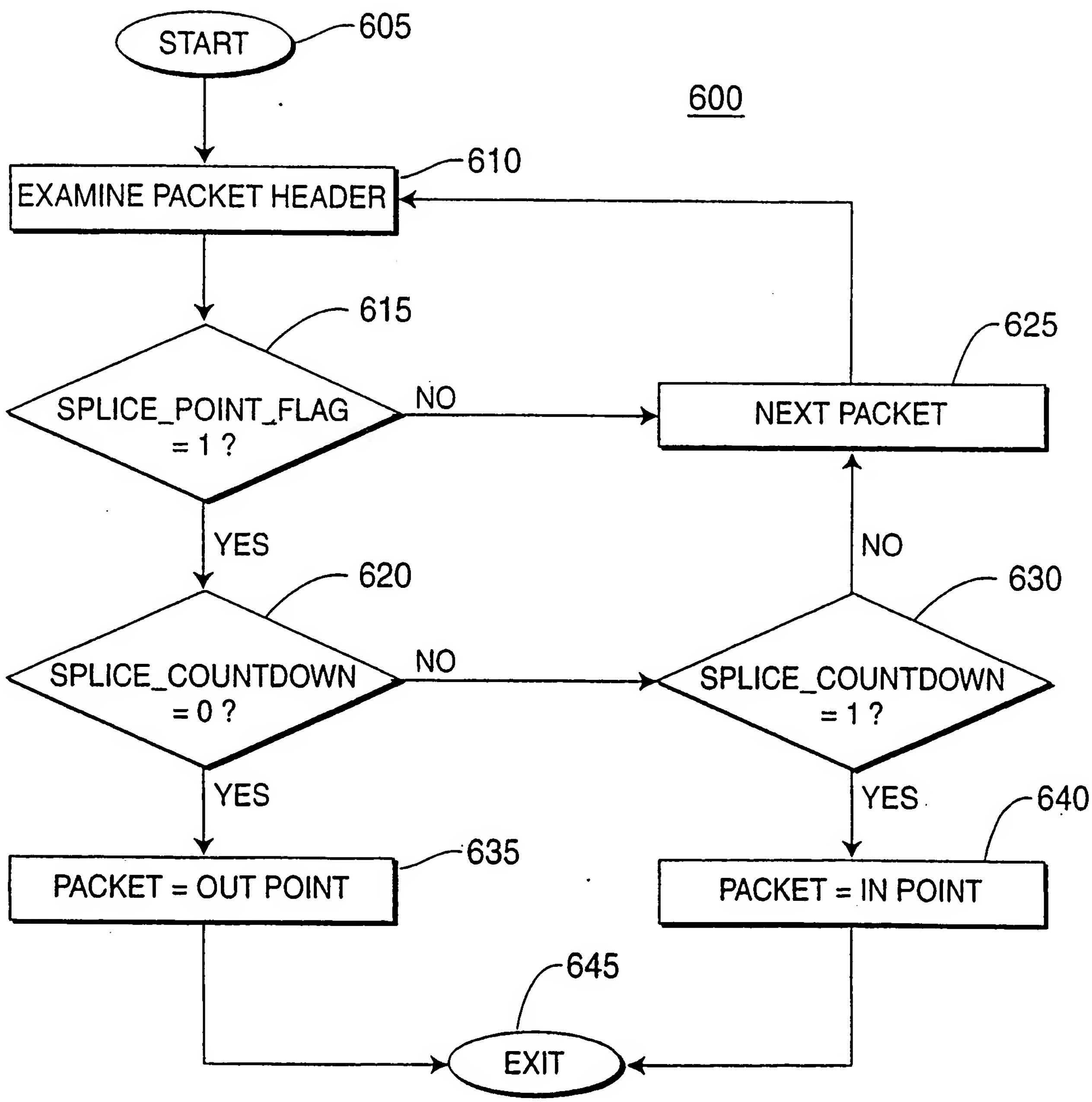

START
605
600
EXAMINE PACKET HEADER
610
615
SPLICE_POINT_FLAG = 1 ?
NO
NEXT PACKET
625
YES
620
SPLICE_COUNTDOWN = 0 ?
NO
630
SPLICE_COUNTDOWN = 1 ?
YES
YES
640
PACKET = OUT POINT
635
PACKET = IN POINT
645
EXIT

FIG. 6